Визуализация графов средствами языка Python и библиотеки igraph

САМОСТОЯТЕЛЬНАЯ РАБОТА

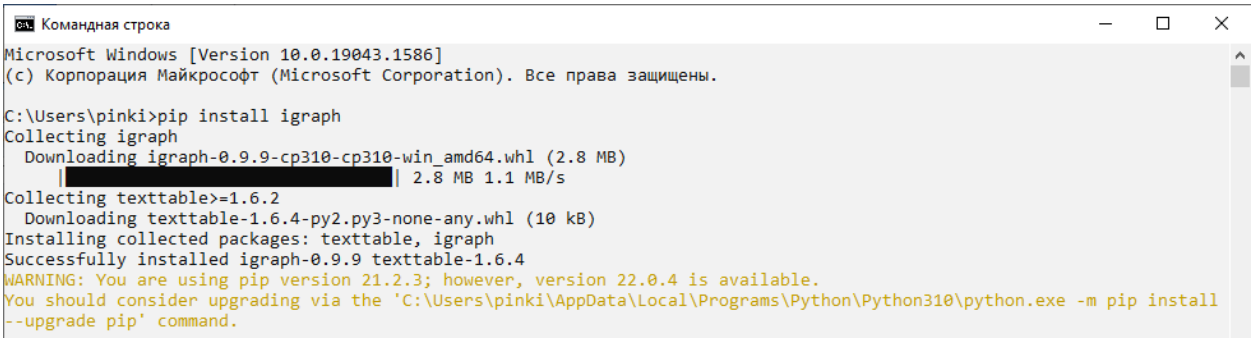
http://www.kimrt.ru

Онлайн-курс

Суперкомпьютерные технологии в задачах моделирования

Библиотека igraph — это набор инструментов сетевого анализа с упором на эффективность, портативность и простоту использования [1].

Установка igraph. Для установки пакета откройте *cmd* и выполните *pip install igraph* и *pip install pycairo* [2]. Пакет *Pycairo* нужен для построения графиков.

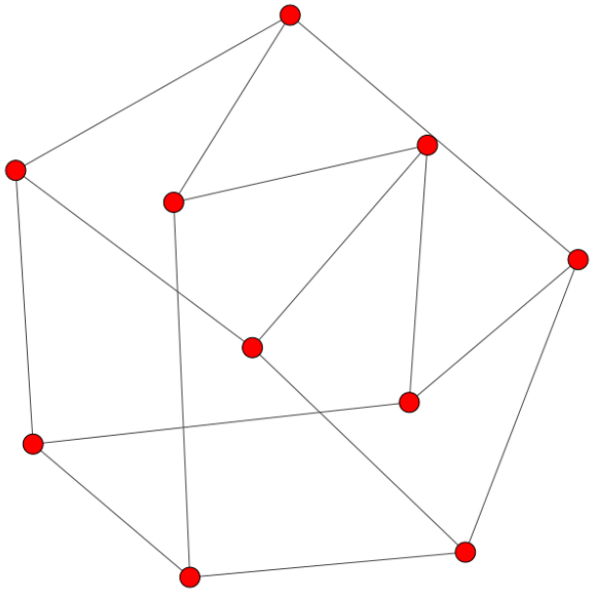


Для проверки правильности установки выполните следующий python скрипт из листинга 1.

Листинг 1

```
import igraph as ig # Импорт библиотеки
g = ig.Graph.Famous("petersen") # Граф Петерсена
ig.plot(g) # Построение графика
```

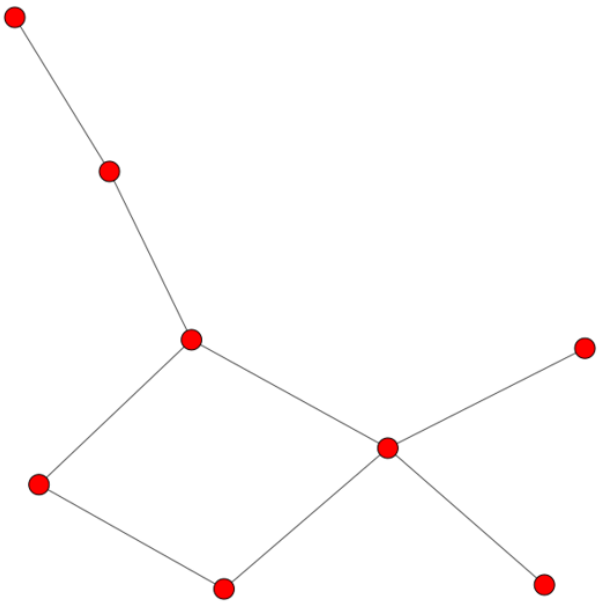
В результате получим представленную ниже картинку графа. Теперь выполним программу из листинга 2.



Листинг 2

```
from igraph import * # импорт библиотеки
g = Graph()  # создание графа
g.add_vertices(8)  # добавление вершин
g.add_edges([(0, 1), (1, 2), (2, 5), (3, 1), (4,
7), (5, 6), (6, 1), (6, 7)]) # добавление ребер
# метод компоновки графа
layout = g.layout_kamada_kawai()
print(g)  # вывод информации о графе в консоль
plot(g, layout=layout)  # построение графика
```

В результате получим следующее изображение графа [3].



Для начала работы с библиотекой *igraph* выполним импорт. Далее нужно создать граф с помощью *Graph()*. Команда *add_vertices()* добавляет вершины графа в количестве, указанном в скобках. С помощью *add_edges()* добавляются ребра графа, в скобках перечисляются связи между вершинами. Ребра и вершины можно удалять с помощью *delete_edges()* и *delete_vertices()* соответственно. Для удаления нужного ребра или вершины необходимо указать индекс элемента, если вы не знаете индекс ребра, воспользуйтесь командой *get_eid()* и укажите в скобках какие вершины связывает нужное вам ребро, например, *get_eid(0,1)*. Можно записать индекс ребра в переменную, вывести на экран или воспользоваться следующей конструкцией для удаления этого ребра.

```
g.delete_edges(g.get_eid(0,1))
```

С помощью *print* можно получить информацию о графе. В первой строке указано количество вершин и ребер, а в третьей указан список ребер.

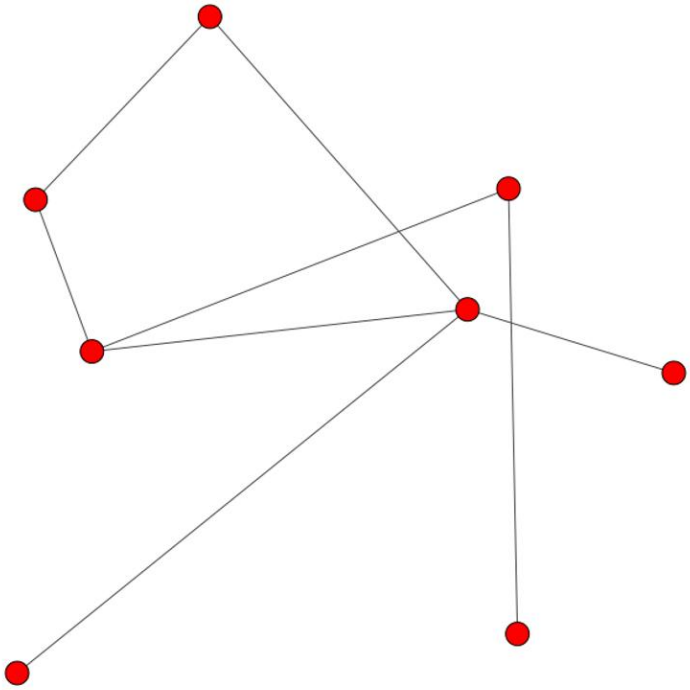
```
IGRAPH U--- 8 8 --
+ edges:
0--1 1--2 2--5 1--3 4--7 5--6 1--6 6--7
```

Для вывода информации без списка ребер существует команда *summary()*. Такой вывод может пригодится при рассмотрении больших графов.

```
IGRAPH U--- 8 8 --
```

С помощью *layout_kamada_kawai()* устанавливается метод компоновки графа необходимый для визуализации в эстетически приятном виде. Команда *plot()* строит наш график.

Если заменить *layout_kamada_kawai()* на *layout_random()* получится подобное изображение.

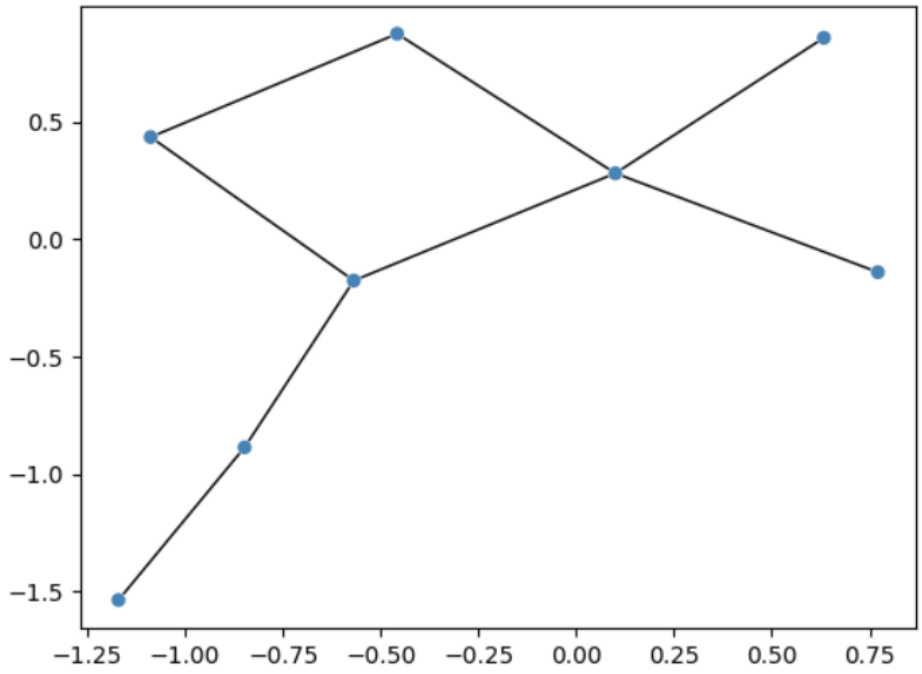


Также возможен вывод изображений графов с помощью библиотеки *matplotlib* (см. листинг 3).

Листинг 3

```
from igraph import * # импорт библиотеки
import matplotlib.pyplot as plt
g = Graph()  # создание графа
g.add_vertices(8)  # добавление вершин
g.add_edges([(0, 1), (1, 2), (2, 5), (3, 1), (4,
7), (5, 6), (6, 1), (6, 7)])  # добавление ребер
print(g)  # вывод информации о графе в консоль
# метод компоновки графа
layout = g.layout_kamada_kawai()
fig, ax = plt.subplots()
plot(g, layout=layout, target=ax)
plt.show()
```

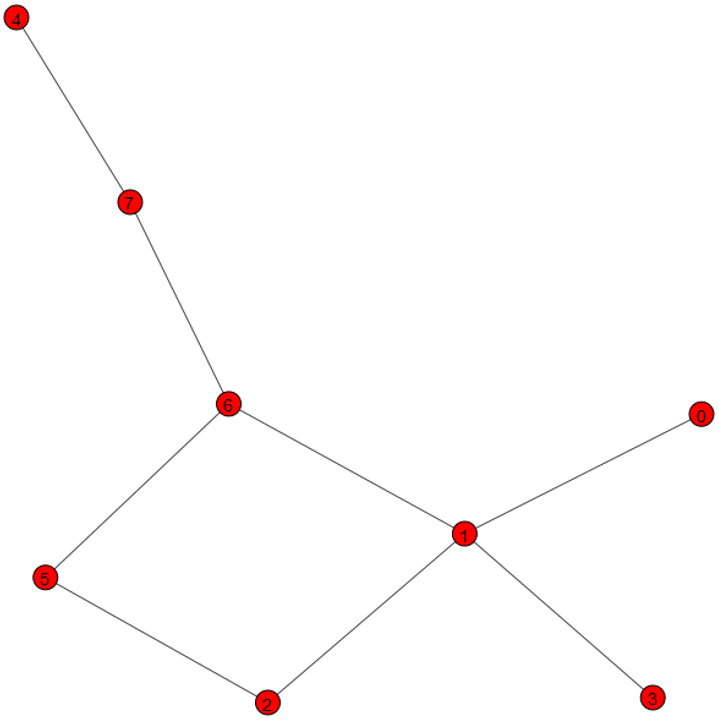
Свойства графика можно изменять различными способами, например, с помощью словаря. В этот словарь можно указывать различные свойства графика: размер, цвет, названия вершин, ширина ребер, метод компоновки и так далее.



Изменим программу из листинга 2, добавив словарь со свойствами, и получил следующую программу (листинг 4). В *vertex_label* указывается название вершины, *range(g.vcount())* перебирает номера вершин [4]. Метод компоновки указывается в *"layout"*, для компоновки выбран тот же метод *kamada_kawai*, но записанный другой командой.

Листинг 4

```
from igraph import *                    # импорт библиотеки
g = Graph()                             # создание графа
g.add_vertices(8)                       # добавление вершин
# добавление ребер
g.add_edges([(0,1),(1,2),(2,5),(3,1),(4,7),(5,6),(
6,1),(6,7)])
# свойства графика
visual_style = {"vertex_label": range(g.vcount()),
"layout": g.layout("kk")}
print(g) # вывод информации о графе в консоль
plot(g, **visual_style) # построение графика
```



http://www.kimrt.ru/index/course_stm/0-24

Проект реализуется победителем Конкурса на предоставление грантов преподавателям магистратуры 2020/2021 благотворительной программы «Стипендиальная программа Владимира Потанина» Благотворительного фонда Владимира Потанина.

Источники

1. Официальный сайт igraph. URL: https://igraph.org/
2. Установка igraph. URL: https://igraph.org/python/tutorial/latest/install.html#installing-igraph
3. Руководство igraph. URL: https://igraph.org/python/tutorial/latest/tutorial.html
4. Getting vertex list from python-igraph. URL: https://stackoverflow.com/questions/30986498/getting-vertex-list-from-python-igraph